OBJECT ORIENTED PROGRAMMING WITH JAVA 8– LAB 11

1. Write a program to create three child threads, one to compute the first 25 even numbers, second to compute the first 50 fibonacci numbers and the third to print the first 20 numbers from the multiplication table of 15 . Implement multithreading using ExecutorService interface in such a way that only 2 threads execute at a time.

Ans=

Code-

```
import java.util.concurrent.*;

class NumbersThread
{
        public static void main(String[] args)
        {
                ExecutorService ec = Executors.newFixedThreadPool(2);

                ec.execute(() -> {
                                    for(int i = 2; i <= 50; i += 2)
                                          {
                                           System.out.println("Even Number: " + i);
                                          }
                           });

                ec.execute(() -> {
                                     int count = 0;
                                     int a = 0;
                                     int b = 1;

                                     while ( count < 25)
                                    {
                                     System.out.println("Fibonacci: " +a);
                                     int fibo = a + b;
                                     a = b;
                                     b = fibo;
                                     count++;
                                    }
                           });

                ec.execute(() -> {
                                    int n = 15;
                                    for(int i = 1; i <= 20; i++)
                                          {
                                           System.out.println("Multiplication table is :" + (n * i));
                                          }
                           });

                ec.shutdown();
        }
}
```

Execution-

Part1,

```
C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>javac NumbersThread.java

C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>java NumbersThread
Even Number: 2
Even Number: 4
Even Number: 6
Even Number: 8
Even Number: 10
Even Number: 12
Even Number: 14
Even Number: 16
Even Number: 18
Even Number: 20
Fibonacci: 0
Fibonacci: 1
Fibonacci: 1
Fibonacci: 2
Fibonacci: 3
Fibonacci: 5
Fibonacci: 8
Fibonacci: 13
Fibonacci: 21
Fibonacci: 34
Fibonacci: 55
Fibonacci: 89
Fibonacci: 144
Fibonacci: 233
Fibonacci: 377
Fibonacci: 610
Fibonacci: 987
Fibonacci: 1597
Fibonacci: 2584
Fibonacci: 4181
Fibonacci: 6765
Fibonacci: 10946
Fibonacci: 17711
Even Number: 22
Fibonacci: 28657
Fibonacci: 46368
```

Part2,

```
Fibonacci: 28657
Fibonacci: 46368
Even Number: 24
Multiplication table is :15
Multiplication table is :30
Multiplication table is :45
Multiplication table is :60
Multiplication table is :75
Multiplication table is :90
Multiplication table is :105
Multiplication table is :120
Multiplication table is :135
Multiplication table is :150
Multiplication table is :165
Multiplication table is :180
Multiplication table is :195
Multiplication table is :210
Multiplication table is :225
Multiplication table is :240
Multiplication table is :255
Multiplication table is :270
Multiplication table is :285
Multiplication table is :300
Even Number: 26
Even Number: 28
Even Number: 30
Even Number: 32
Even Number: 34
Even Number: 36
Even Number: 38
Even Number: 40
Even Number: 42
Even Number: 44
Even Number: 46
Even Number: 48
Even Number: 50
```

2. Write a program to extract a portion of an ArrayList.

Ans=

Code-

```
import java.util.List;
import java.util.ArrayList;

class ArrayFruitList
{
        public static void main(String[] args)
        {
                ArrayList<String> FruitsList = new ArrayList<>();

                FruitsList.add("Apple");
                FruitsList.add("Mango");
                FruitsList.add("Orange");
                FruitsList.add("Grapes");
                FruitsList.add("Banana");
                System.out.println("Fruits List: " + FruitsList);

                List<String> ExtractedFruits = FruitsList.subList(1,4);
                System.out.println("Extracted Portion: " + ExtractedFruits);
        }
}
```

Execution-

```
C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>javac ArrayFruitList.java

C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>java ArrayFruitList
Fruits List: [Apple, Mango, Orange, Grapes, Banana]
Extracted Portion: [Mango, Orange, Grapes]

C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>
```

3. Write a program to update a specific element in an ArrayList by a given element and remove the last element from the ArrayList and print it using for each loop.

Ans=

Code-

```
import java.util.List;
import java.util.ArrayList;
import java.util.Iterator;

class ArrayAnimalList
{
        public static void main(String[] args)
        {
                ArrayList<String> AnimalList = new ArrayList<>();

                AnimalList.add("Tiger");
                AnimalList.add("Lion");
                AnimalList.add("Elephant");
                AnimalList.add("Monkey");
                AnimalList.add("Leopard");
                System.out.println("Animals :" + AnimalList);

                AnimalList.set(2,"Wolf");
                AnimalList.set(4,"Deer");
                System.out.println("Animals after updating: " + AnimalList);

                Iterator<String> iterator = AnimalList.iterator();
                String lastElement = null;
                while (iterator.hasNext())
                {
                 lastElement = iterator.next();
                 if (!iterator.hasNext())
                        {
                         iterator.remove();
                        }
                }

                System.out.println("Removed Last Animal: " + lastElement);
                System.out.println("Final Animal list: " + AnimalList);
        }
}
```

Execution-

```
C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>javac ArrayAnimalList.java

C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>java ArrayAnimalList
Animals :[Tiger, Lion, Elephant, Monkey, Leopard]
Animals after updating: [Tiger, Lion, Wolf, Monkey, Deer]
Removed Last Animal: Deer
Final Animal list: [Tiger, Lion, Wolf, Monkey]
```

4. Create a class named Student with following members

a. Data Members

i. regno

ii. name

iii. marks

b. Constructor to accept all values

c. Method to print the values of data members

Create another class named AddStudent which contains main method to create

Student objects and add it to an ArrayList and display the details of all the students using ListIterator.

Ans=

Code-

```
class Student
{
        int regno;
        String name;
        int marks;

        public Student( int regno, String name, int marks)
        {
                this.regno = regno;
                this.name = name;
                this.marks = marks;
        }

        public void studentDetails()
        {
                System.out.println("Registration Number: " + regno);
                System.out.println("Student Name: " + name);
                System.out.println("Student Marks: " + marks);
        }
}

public class AddStudent
{       public static void main(String[] args)
        {
                ArrayList<Student> studentList = new ArrayList<>();

                studentList.add(new Student(1, "Sunil", 95));
                studentList.add(new Student(2, "Rushi", 92));
                studentList.add(new Student(3, "Yogesh",85));
                studentList.add(new Student(4, "Ramesh",82));
                studentList.add(new Student(5, "Suresh",96));

                ListIterator<Student>iterator = studentList.listIterator();
                while (iterator.hasNext())
                        {
                         Student S = iterator.next();
                         S.studentDetails();
                         System.out.println();
                        }
        }
}
```

Execution-

```
C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>java AddStudent
Registration Number: 1
Student Name: Sunil
Student Marks: 95

Registration Number: 2
Student Name: Rushi
Student Marks: 92

Registration Number: 3
Student Name: Yogesh
Student Marks: 85

Registration Number: 4
Student Name: Ramesh
Student Marks: 82

Registration Number: 5
Student Name: Suresh
Student Marks: 96


C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>
```

5. Create a TreeSet and add some Integer objects to it. Create another TreeSet and copy only those elements from first TreeSet to the second which are greater than or equal to a specified value given as input. Display the second TreeSet values as output using Iterator.

Ans=

Code-

```
import java.util.Scanner;
import java.util.TreeSet;
import java.util.Iterator;

class TreeSetMain
{
        public static void main(String[] args)
        {
                TreeSet<Integer> ts1 = new TreeSet<>();

                ts1.add(1);
                ts1.add(5);
                ts1.add(10);
                ts1.add(15);
                ts1.add(20);

                TreeSet<Integer> ts2 = new TreeSet<>();

                Scanner P = new Scanner(System.in);
                System.out.print("Enter a value: ");
                int enteredValue = P.nextInt();

                 for (Integer n : ts1)
                {
                        if (n >= enteredValue)
                                {
                                 ts2.add(n);
                                }
                }
        System.out.println("Elements in the second TreeSet greater than or equal to " + enteredValue + ":");
        Iterator<Integer> iterator = ts2.iterator();
        while (iterator.hasNext())
                {
                 System.out.println(iterator.next());
                }
        }
}
```

Execution-

```
C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>javac TreeSetMain.java

C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>java TreeSetMain
Enter a value: 7
Elements in the second TreeSet greater than or equal to 7:
10
15
20

C:\Users\p7pha\OneDrive\Desktop\Cdac DBDA\JAVA>
```